

Prior Art

Fig. 1

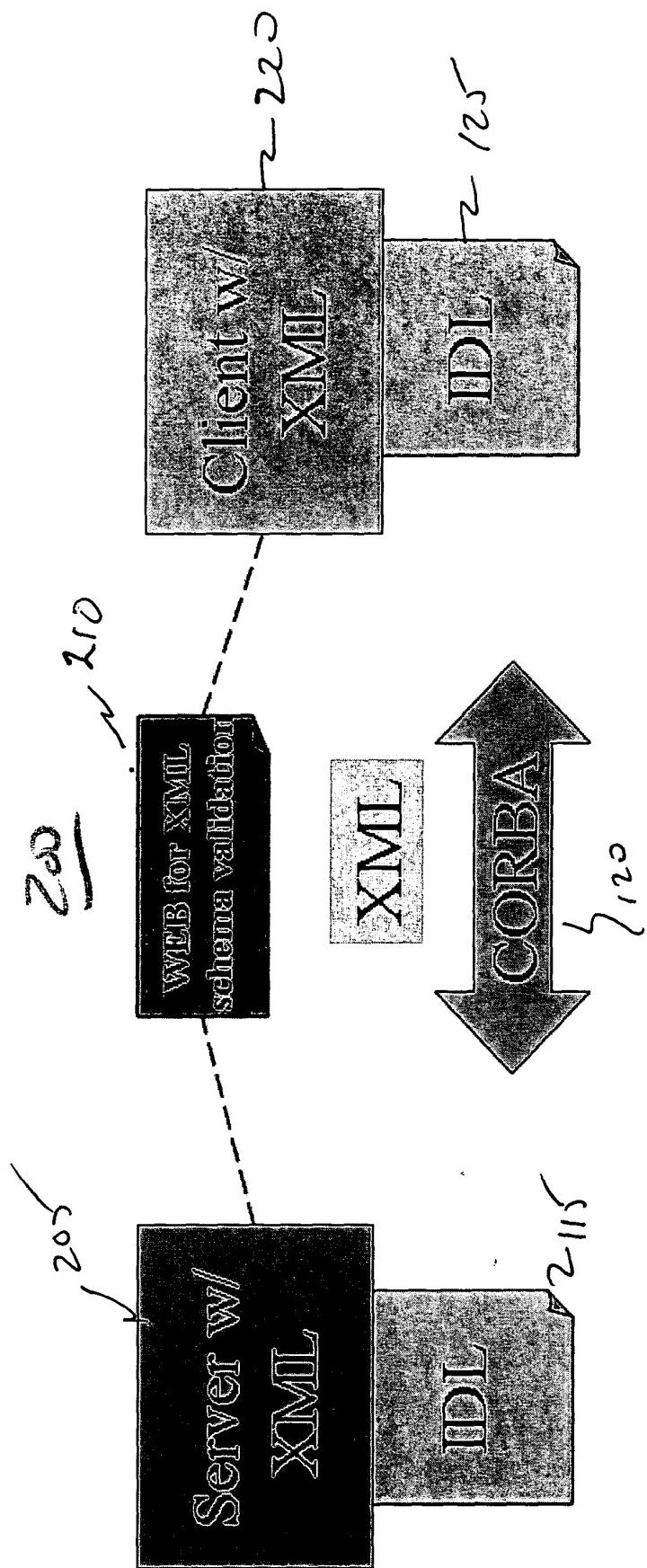
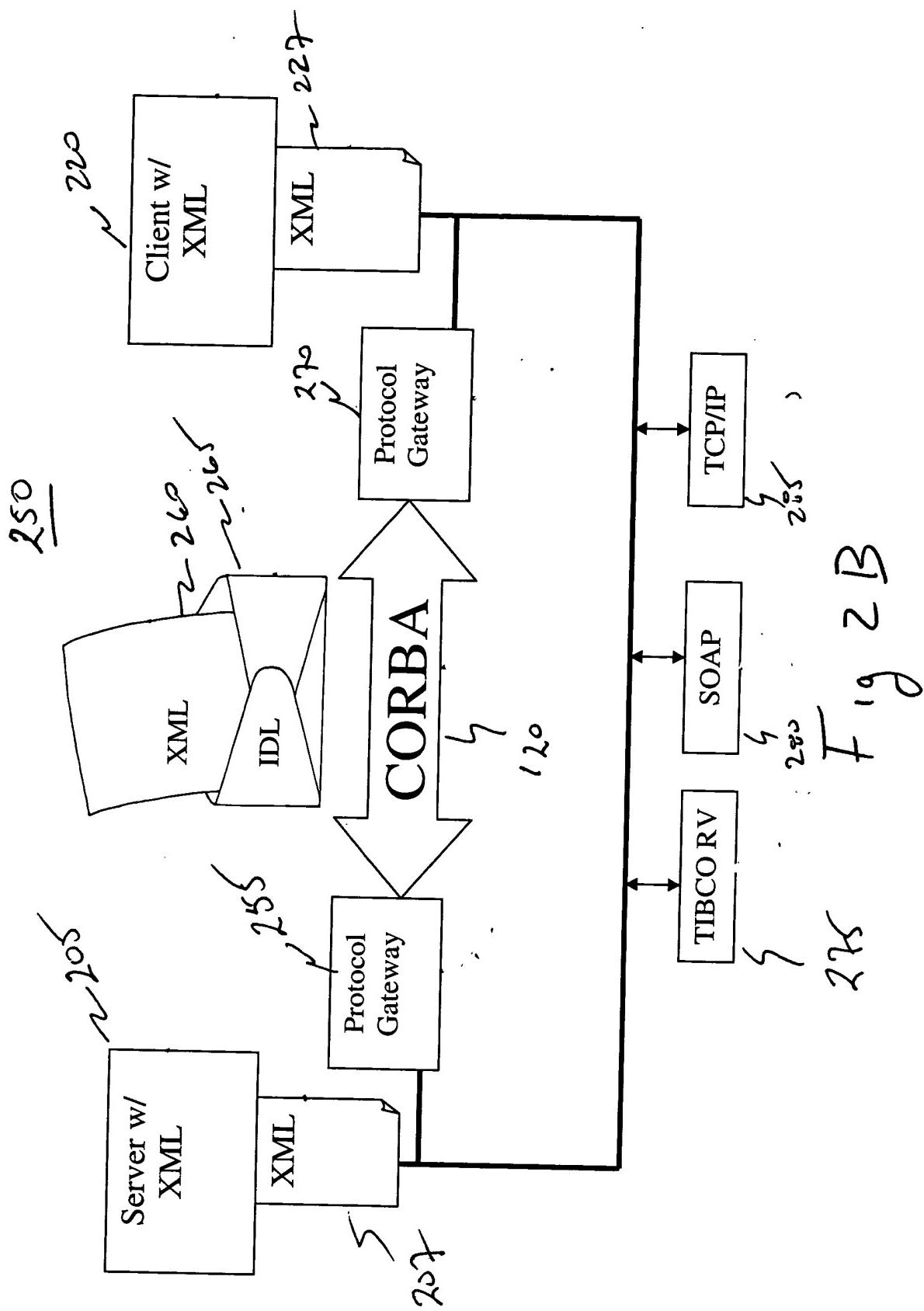


Fig. 2A



```
Module MES_Main{
    Interface Transaction_Tempalte{
        String Siview_Transaction(in String XML_Argv);
    }
}
```

Fig. 3

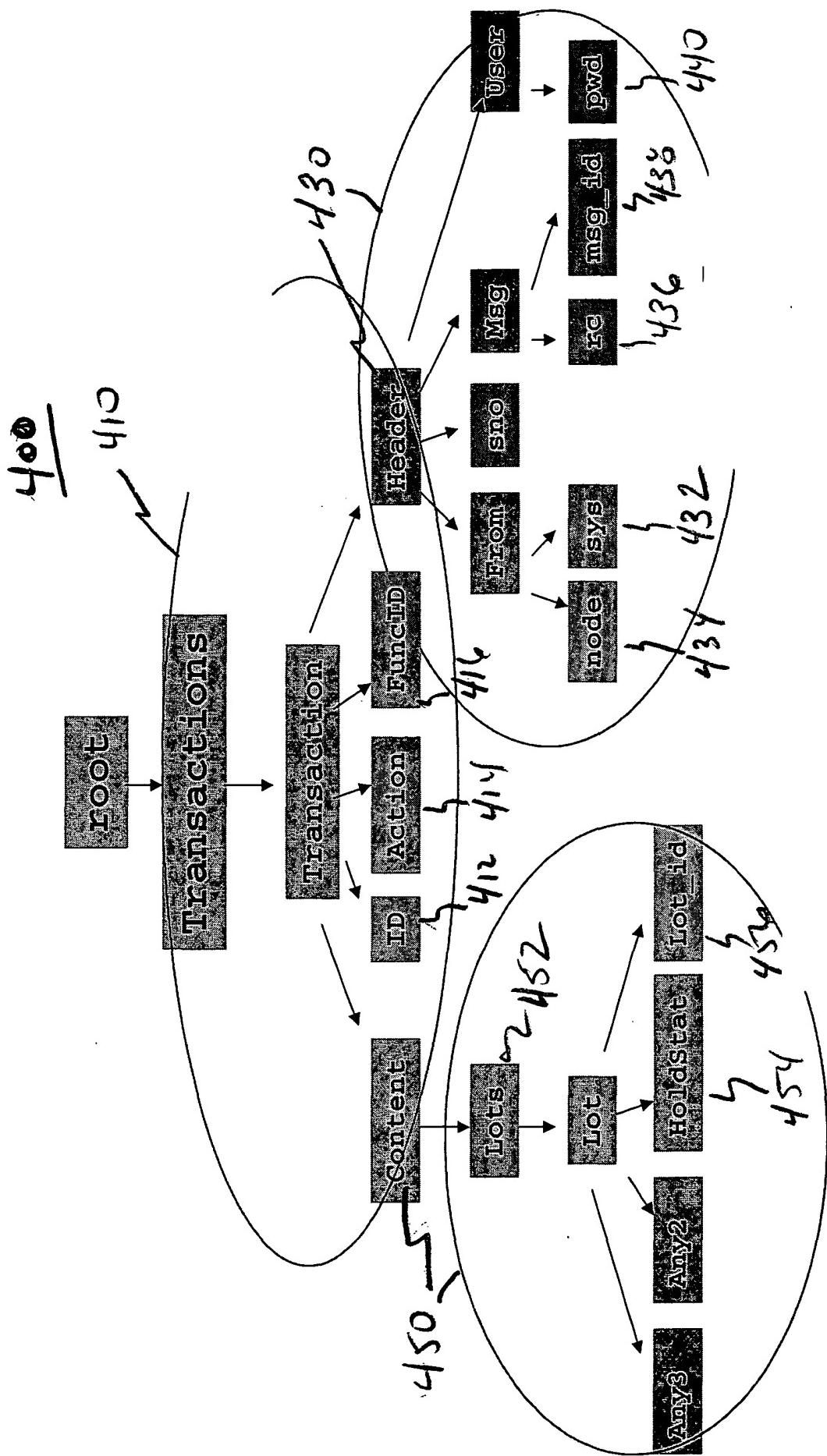


FIGURE 4

410

D:\Data\syncos\testSchema sample TXT

```
83 // define how transaction looks like line 87~101
84 // every transaction has 3 attributes for id name, action type = { Inquiry or ResultReply}, function id
85 // and 2 elements header & Contents
86 // these attributes are filled out by client or server
87
88 <AttributeType name="ID" required="yes" dt:type="string"/> ~~~~~ 510
89 <AttributeType name="Action" dt:type="string" required="yes"/> ~~~~~ 512
90 <AttributeType name="FuncID" dt:type="string" required="yes"/> ~~~~~ 514
91 <ElementType name="Transaction" content="mixed">
92   <attribute type="ID"/> ~~~~~ 510
93   <attribute type="Action"/> ~~~~~ 512
94   <attribute type="FuncID"/> ~~~~~ 514
95   <element type="Header"/> ~~~~~ 516
96   <element type="Contents"/> ~~~~~ 520
97 </ElementType> ~~~~~ 540
98
99 // transactions may include many transaction
100 <ElementType name="Transactions" content="eltOnly"> } -410
101   <element type="Transaction"/>
102 </ElementType>
103
104 </Schema>
105
```

Figure 5'a

4/30

```

46 // define header line 50~80
47 // Message has 2 attributes and message data
48 // 1. return code = rc
49 // 2. message id = msg_id
50 <AttributeType name="rc" required="yes" dt:type="string"/> ~ 432
51 <AttributeType name="msg_id" required="yes" dt:type="string"/> ~ 431
52 <ElementType name="Msg" content="eltOnly">
53 <attribute type="rc"/>
54 <attribute type="msg_id"/>
55 </ElementType>

56 // 2 attributes here to indicate "from which client or server node = node and system = sys"
57 <AttributeType name="node" required="yes" dt:type="string"/> ~ 440
58 <AttributeType name="sys" required="yes" dt:type="string"/> ~ 440
59 <ElementType name="From" content="eltOnly"> ~ 442
60 <attribute type="node"/> ~ 440
61 <attribute type="sys"/> ~ 442
62 </ElementType> ~ 442

63 // user has password => pwd as its attribute
64 <AttributeType name="pwd" required="yes" dt:type="string"/> ~ 446
65 <ElementType name="User" content="eltOnly"> ~ 446
66 <attribute type="pwd"/> ~ 446
67 </ElementType> ~ 446

68 // define header line 50~80
69 // header has one attribute called serial number => sno
70 // and may have 3 elements User, From & Msg=>Message
71 <AttributeType name="sno" required="yes" dt:type="string"/> ~ 448
72 <ElementType name="Header" content="mixed">
73 <attribute type="sno"/> ~ 448
74 <element type="From"/> ~ 438
75 <element type="Msg"/> ~ 438
76 <element type="User"/> ~ 438
77 </ElementType> ~ 444
78
79
80
81

```

FIGURE 56

15 // define lot features line 10 ~ 38
16 // lot has one attribute named lot_id for replying result to client
17 // and many other features in terms of elements
18 // ex: hold state
19
20 <AttributeType name="txt" required="yes" dt:type="string"/> ~460
21 <ElementType name="HoldState" content="eltOnly"> { 458
22 </ElementType>
23
24 <ElementType name="any2" content="eltOnly"> { 456
25 <attribute type="txt"/> ~460 !
26 </ElementType>
27
28 <ElementType name="any3" content="eltOnly"> { ~454
29 <attribute type="txt"/> ~460 !
30 </ElementType>
31
32 <ElementType name="Lot" content="mixed">
33 <attribute type="Lot_id"/> ~460 !
34 <element type="HoldState"/> ~458 !
35 <element type="any2"/> ~456 !
36 <element type="any3"/> ~454 !
37 </ElementType>
38
39 <ElementType name="Lots" content="eltOnly"> { 450 !
40 <element type="Lot"/> ~452 !
41 </ElementType>

Figure 5c

660

```
<?xml version="1.0" ?>
<Transactions xmlns="x-schema:http://SomeWebServer/TestSchema.xml">
    // transactions may include many transaction
    // this XML has a schema file located at http://SomeWebServer
    <Transaction id="LotInfoInq" Action="Inquiry" FuncId="0001">          615
        // every transaction has 3 attributes for id name, action type = {Inquiry or ResultReply}, function id
        // and 2 elements header & Contents
        // these attributes are filled out by client here
            <Header sno="00100">          622
                // header has one attribute called serial number
                // and 2 elements From & User
                    <From node="MyPC" sys="OMI" />          624
                    // 2 attributes here to indicate "from which client node = node and system = sys"
                    <User pwd="123">ABC</User>          625
                    // user "ABC" has password "123" as its attribute
            </Header>
            <Contents>
                // contents may have many lots as it content
                    <Lots>          632
                        // one of its lot
                        <Lot lot_id="ABC100.00"> // lot has one attribute named lot_id for client input
                            <any2 txt="1" />          // and other necessary input argument, if any
                            <any3 txt="2" />          // and other necessary input argument, if any
                        </Lot>
                    </Lots>
                </Contents>
            </Transactions>
```

FIGURE 601

XML Version="1.0"?>

```
<Transactions xmlns="x-schema: http://SomeWebServer/TestSchema.xml">
    // transactions may include many transaction
    // this XML has a schema file located at http://SomeWebServer
    <Transaction ID="LotInfoInqReply" Action="ResultReply" FuncID="0005"> #15
        // every transaction has 3 attributes for id name, action type = {Inquiry or ResultReply}, function id
        // and 2 elements header & Contents
        // these attributes are filled out by server here

        <Header sno="00101">
            // header has one attribute called serial number
            // and 2 elements From & Msg-Message
            <From node="MesServer" sys="MMS" /> #16
                // 2 attributes here to indicate "from which server node = node and system = sys"
                <Msg rc="0" msg_id="0001">Normal End</Msg> #17
                    // Message has 2 attributes and message data => here is "Normal End"
                    // 1. return code = rc
                    // 2. message id = msg_id

            </Header>
            <Contents>
                // contents may have many lots as its result #18
                <Lots>
                    // one of its lot information result #19
                    <Lot lot_id="ABC100.00">
                        // lot has one attribute named lot_id for replying result to client
                        // and many other results in terms of elements
                        <HoldState txt="NotOnHold"/> #20
                            // for example: HoldState result with text data = "NotOnHold"
                            <any2 txt="reply stuff..."/>
                            // other result, if any
                            <any3 txt="reply stuff..."/>
                            // other result, if any #21
                    </Lot>
                </Lots>
            </Contents>
        <Transaction>
            // contents may have many lots as its result #22
            <Lots>
                // one of its lot information result #23
                <Lot lot_id="ABC100.00">
                    // lot has one attribute named lot_id for replying result to client
                    // and many other results in terms of elements
                    <HoldState txt="NotOnHold"/> #24
                        // for example: HoldState result with text data = "NotOnHold"
                        <any2 txt="reply stuff..."/>
                        // other result, if any
                        <any3 txt="reply stuff..."/>
                        // other result, if any #25
                </Lot>
            </Lots>
        </Transaction>
    </Transactions>
```